

# ACELP/MP-MLQ에 기초한 dual-rate 음성 코더의 DSP 구현

## Implementation of the ACELP/MPMLQ-Based Dual-Rate Voice Coder Using DSP

이 재 식\*, 손 용 기\*, 전 일\*, 장 태 규\*, 민 병 기\*\*

\*중앙대학교 전자전기공학부, \*\*한국전자통신연구원

Jae-Sik Lee\*, Yong-Ki Son\*, Il Jeon\*, Tae-Gyu Chang\*, Byoung-Ki Min\*\*

\*School of Electronics and Electrical Engineering, Chung-Ang University

\*\*Electronics and Telecommunications Research Institute

\*tgchang@jupiter.cie.cau.ac.kr, \*\*min@computer.etri.re.kr

### Abstract

This paper describes the fixed-point DSP implementation of a CELP(code-excited linear prediction)-based speech coder. The effective realization methodologies to maximize the utilization of the DSP's architectural features, specifically parallel movement and pipelining are also presented together with the implementation results targeted for the ITU-T standard G.723.1 using Motorola DSP56309. The operation of the implemented speech coder is verified using the test vectors offered by the standard as well as using the peripheral interface circuits designed for the coder's real-time operation.

### 1. 서 론

압축된 음성 신호의 전송은 인터넷 및 이동 통신 등을 통한 멀티미디어 서비스에 있어서 제한된 대역폭을 효율적으로 사용토록 하여주는 핵심기술로서 이에 대한 연구 및 적용 기술에 대한 연구가 매우 활발하게 진행되고 있다. 특히 최근 들어 digital hardware 기술의 급속한 발전에 힘입어 종래에는 프로세서의 처리속도 부담 때문에 응용이 제한되었던 analysis-by-synthesis에 의한 음성압축 기술의 적용이 활발하게 확대되고 있는 추세이다.[1][2][3] Analysis-by-synthesis 기법은 압축과정에서 미리 음성신호를 합성하여 실제 입력신호와의 차이를 구하여 보고 이것이 최소가 되도록 하여 주는 압축 변수 값들을 탐색에 기초하여 얻는

방법이다. 이러한 방식에서 excitation parameter 및 pitch parameter들을 탐색에 의해 구하는 과정은 프로세서의 많은 처리 부담을 요구하게 된다.

음성압축 알고리즘은 디지털 필터링, recursion등을 포함한 수치적인 연산의 비중이 크기 때문에 DSP(digital signal processor)를 이용하여 구현하는 것이 효과적이다. 특히 DSP는 알고리즘을 고속으로 처리하도록 parallel move, 명령어 pipelining 및 특수 hardware 구조를 바탕으로 하는 instruction 기능들을 제공하는 것이 일반적이다. 따라서 이러한 DSP의 기능적인 특성을 고려한 알고리즘 최적화는 상대적으로 수치적 연산이 많고 복잡도가 큰 저비트율 음성압축 알고리즘을 구현하는데 있어서 매우 중요한 사항이라 하겠다. 이에 본 논문에서는 저비트율 음성 코더 중 최근 주류를 이루고 있는 CELP방식을 대상으로 이를 구성하는 각각의 기능 블록들의 동작원리 및 DSP 구현 구조에 대하여 살펴보고, 이를 Motorola 24-bit DSP를 이용하여 ITU-T G.723.1 [5]의 표준에 따라 구현한 음성코더에 관하여 기술하였다.

### 2. CELP에 기초한 음성 코더의 구성 및 동작 원리

포먼트(formant) 합성필터와 피치(pitch) 합성필터의 cascade된 조합에 여기펄스(excitation pulse)를 가함으로써 음성을 합성할 수 있다는 것이 CELP를 기반으로 하는 음성

압축에서 따르는 기본적인 원리이다.[2][3] 따라서 입력된 음성 신호의 구조적 특성을 short-term 성분, long-term 성분, 여기(excitation) 신호의 세 부분으로 나누고 각 성분들을 linear-prediction 기법과 iterative code-book search 기법을 통하여 각각에 해당하는 포먼트 합성필터 계수(LPC, linear-prediction coefficient), 피치 합성필터 계수, 여기펄스 정보를 구하고 이의 색인을 전송한다. 그림 1에 CELP에 기초한 음성압축 시스템의 기본적인 구성도를 나타내었다.

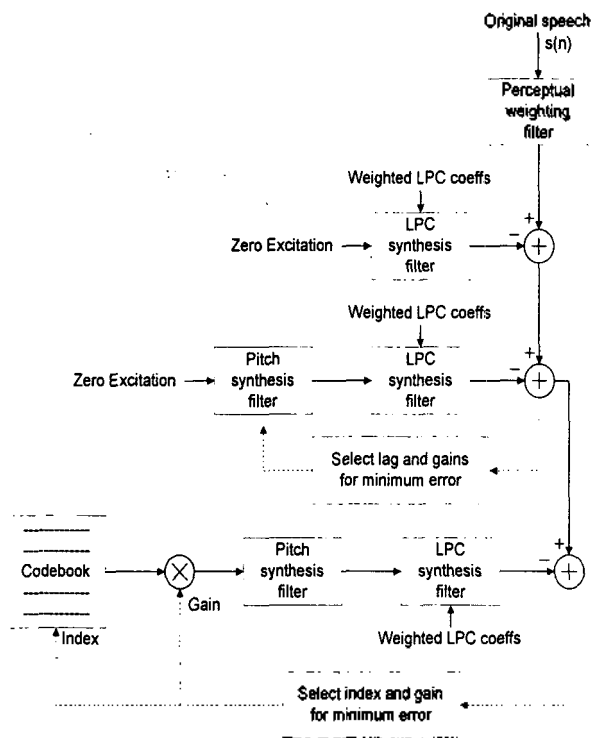


그림 1. CELP에 기초한 음성 코더의 기본적 구성도

포먼트 합성필터의 계수, 즉 LPC를 구하는 것은 입력신호의 correlation값들을 계산하여 Levinson-Durbin 알고리즘 등과 같은 recursive 알고리즘에 의하여 구하는 것이 일반적이다. 그림 1의 하단에서 포먼트 합성필터와 피치 합성필터와 여기펄스의 조합에 의해 음성을 합성하는 것을 보이고 있는데, 이는 압축단에서 음성신호를 합성하여 기준이 되는 입력 음성신호와 오차를 구하고 이를 최소화 하는 성분의 인덱스를 찾는 과정을 갖는다. 특히 음질에 가장 큰 영향을 미치는 여기벡터를 찾는 과정의 비중을 높게 하여 탐색을 하게 된다. 이러한 기법은 압축단에서 분석한 성분들을 이용하여 합성하는 과정이 없는 analysis-and-synthesis 방식과 구별하여 analysis-by-synthesis 기법이라 칭한다. 또한 음성을 합성하는 과정에 사람의 청각특성(psychoacoustics)을 반영

한 양자화 잡음을 분포시키기 위한 목적으로 인지가중필터(perceptual weighting filter)를 적용함으로써 동일 압축률에 대해 음질을 올리는 효과를 얻는다. 피치 추정은 일반적으로 음성의 자기상관값을 이용한 open-loop 피치 추정 및 순환방식에 의한 closed-loop 피치추정 방식을 복합적으로 사용하여 처리 효율을 증대시키고 있다. 여기벡터들은 미리 코드북 형태로 저장해 놓고, 해당하는 색인으로 여기신호를 표현할 수 있도록 한다. 적합한 여기벡터를 찾는 과정은 사전에 구한 포먼트 합성 필터와 피치 합성 필터의 계수를 이용하여 음성신호를 합성하고, 합성한 음성신호와 과거에 의한 영향이 제거된 음성과의 오차를 최소로 하는 코드북 색인을 구한다. 이와 같은 포먼트 합성필터 계수, 피치 합성필터 계수, 및 여기벡터로 음성을 표현하고 전송함으로써 음성을 압축하는 것이 CELP방식의 기본 원리이다.

### 3. 음성 코더의 DSP 구현

본 논제에서는 ITU-T G.723.1의 표준에 따른 음성코더의 구조에 대하여 기술하고 Motorola DSP 56309를 이용하여 구현한 결과를 처리성능을 제고하기 위한 설계고려사항 및 구현 방법들과 함께 기술하였다.

G.723.1은 화상전화 서비스를 위한 음성 압축 표준을 주목적으로 ITU-T에서 1995년에 확정하였으며, ACELP(algebraic-code-excited linear prediction) 및 MP-MLQ(multi-pulse maximum likelihood quantization) 방식으로 각각 5.3/6.3 kbit/sec의 dual rate 지원 구조를 갖고 있다.[5] 처리 frame 단위는 30 msec 이고 입력력 data는 8 KHz sampling rate에 16bit linear PCM format 이다. 이 표준은 bit exact한 규정 이므로 표준에서 제공된 test pattern들에 대한 시험을 통하여 올바른 구현을 확인하여 볼 수 있도록 하였다.

그림 2에 음성 부호기의 기본 구조를 나타내었다. 전처리부에서는 입력신호를 받아 frame 단위로 buffering을 수행하는 부분과 입력신호의 DC성분을 제거하기 위한 highpass 필터를 포함하고 있다. 입력신호는 30msec 단위로 240 sample씩 buffering하고 buffer를 두 개를 두어 switching에 의해 처리중인 data와 입력을 받고있는 buffer를 구분하도록 하였다. Timer-interrupt에 의해 8KHz로 입력하도록 하였다. 필터는 127/128 위치에 하나의 극점을 갖는 IIR(infinite-impulse response) 필터로 구현하였다.

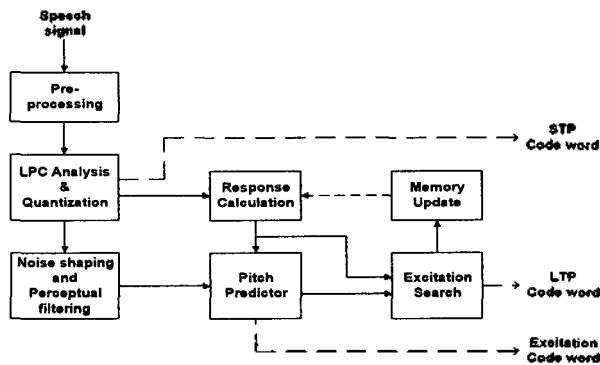


그림 2. G.723.1의 부호기 기능 블록 구성도

인지가중필터와 잡음형상 필터는 입력신호의 에너지가 많이 몰려있는 주파수 대역에서 gain을 줄여줌으로서 오차의 영향을 계산하는 과정에서 상대적인 step size를 키워주는 효과를 주어 궁극적으로는 신호 대 양자화 잡음비를 향상시키는 역할을 수행한다. 포먼트 합성필터는 10차 AR(auto-regressive) 모델로 나타내었으며 LPC를 구하는 것은 한 frame 즉 240 sample을 4개의 부프레임으로 나누고 각각의 부프레임에 대해 Levinson-Durbin 기법을 적용하여 구했다. 마지막 부프레임의 LPC값은 PSVQ(predictive split vector quantizer)를 이용하여 양자화 시킨 후 전송한다. 피치 측정부에서는 2개의 부프레임 데이터에 해당하는 120샘플에 대한 피치를 구하는데, 데이터의 correlation을 계산하여 구한다. 피치 예측부에서는 60샘플의 데이터에 대해 이미 구한 피치값을 대상으로 자세한 피치값을 구한다.

ACELP 방식의 경우 여기신호로 4개의 여기펄스를 찾게 되는데 각 펄스는 미리 고정된 8개의 위치 중의 한 위치를 가지게 된다. 펄스의 위치와 부호가 정해지면, 전체의 펄스 위치를 1 grid 만큼을 이동하여 얻은 신호와 이동하지 않았을 때 재생한 신호 중 더 적합한 신호를 취하여 그 정보를 코드화하여 전송한다. 펄스의 위치를 탐색하기 위해서는 총 4096(8×8×8×8)회의 loop중 탐색 부담을 줄이기 위해서, 새 개의 펄스 위치를 바꾸어 비교할 값을 구하여 미리 계산한 문턱값(threshold)과 비교하여 조건이 만족할 때에만 나머지 한 펄스에 대한 비교를 150회 내에서 실시한다.

DSP56309는 오디오 응용 목적으로 제공된 24bit DSP 프로세서로서 pipeline에 의한 명령어 처리 구조와 함께 100MIPS에 달하는 처리속도를 가지고, 프로그램 메모리와 x,y 데이터 메모리를 합쳐 20Kword에 해당하는 내부 메모리를 가지고 있다. 기본적으로 24bit 단위로 처리하게 되어 있지만, 프로세서 내부의 레지스터 설정으로 16bit 단위의 처리

도 가능하게 되어 있으므로 16bit 기반으로 구성된 G.723.1 표준을 구현할 수 있다. 프로세서를 프로그램 메모리와 데이터 메모리를 동시에 접근할 수 있는 parallel move 기능을 활용하면 명령어 수행과 데이터 이동을 한 명령어로 수행할 수 있게 하여 처리시간을 줄일 수 있게 된다. 특히 DSP56309계열의 DSP는 x,y 두 그룹의 데이터를 가지고 있는데, 두 그룹 데이터를 동시에 접근할 수 있어서 한 번에 2개의 데이터를 이동할 수 있는 장점을 가진다. 음성 코덱의 code 탐색부, 필터링부, correlation부 등에서는 2개의 데이터를 동시에 접근하면서 반복적으로 수행하는데, 이는 세 번의 동작을 한 명령어로 처리하므로, 3배에 달하는 처리효율 향상을 도모할 수 있다.

DSP가 가지는 pipeline구조에 의해 모든 명령어를 1클럭에 처리하는 효과를 얻게 된다. 따라서 pipeline 구조에 위배되는 조합의 명령어 순서를 가지는 경우는 명령어 순서를 재정렬하는 방법을 이용하여 처리 효율을 줄이지 않도록 하였다. 전형적인 pipeline stall을 가져오는 경우는 accumulator의 값을 변경하는 명령어 다음에 accumulator값을 읽으려는 명령어를 쓰거나, 상태레지스터 값을 읽으려는 명령어를 사용하는 경우이다. DSP 처리 효율을 높여주는데 필요한 DSP56309계열의 특수 명령어로는 MAX, CLB, NORMF등이 있다. 두 값을 비교하여 큰 값을 되돌리는 동작을 수행하는 MAX라는 명령어는 비교, 분기, 값 이동의 새 명령어를 단 하나의 명령어로 수행할 수 있다. CLB와 NORMF 명령어는 연산과정의 데이터 resolution을 높여주기 위한 목적으로, 정규화할 데이터의 비트수를 탐색하는 부분과 탐색한 비트수만큼을 shift하는 연산을 2명령어로 해결할 수 있으므로 처리속도 향상에 기여할 수 있다.

#### 4. 실험 및 결과

표 1에 DSP로 구현한 음성코덱의 각 부분별 소요 MIPS와 memory 소요량을 요약하였다. Pipeline stall된 것까지 합하면 encoder가 26.6 MIPS이며 memory 소요량은 약 7 Kwords에 달함을 알 수 있다. Decoder는 MIPS 소요량이 작아 encoder와 합칠 경우 30.8 MIPS에 이르며 memory 소요량에는 거의 변화가 없다. MP-MLQ 방식의 6.3 kbit/sec 전송률의 경우는 MIPS 소요량이 5.3 kbit/sec의 ACELP 방식에 비해 약 0.9 MIPS가 더 소요되며 memory 소요량도 약 1 Kword 더 필요하다.

표 1. ACELP에 의한 음성코더 DSP 구현 MIPS 및 메모리 소요량 요약

각 블록별 기능	MIPS	Memory table(x)	Memory table(y)
<b>Encoder total</b>	<b>26.6</b>	<b>3,610</b>	<b>3,438</b>
LPC analysis	0.91	200	-
LSP quantization	2.9	10	3,072
Pitch search	9.1	3,400	-
ACELP search	9.3	-	364
<b>Decoder total</b>	<b>2.3</b>	<b>170</b>	<b>-</b>
Data buffering	0.2	-	-
Data I/O	1.7	-	-
<b>Total</b>	<b>30.80</b>	<b>3,780</b>	<b>3,438</b>

### 5. 동작시험

본 연구에서 구현한 음성코더의 동작을 시험하기 위하여 그림 3과 같이 DSP56309를 기반으로 시험환경을 구성하였다. 실제 음성신호를 필터링하여 DSP의 입력으로 들어오면, 8 KHz로 A/D 변환하여 16비트 PCM으로 데이터가 입력되어 코더에 의해 인코딩이 되고, 디코더 측에서는 압축된 음성에 대응하는 데이터를 디코딩하여 음성을 재조합해 스피커로 실시간으로 소리를 들을 수 있는 환경을 구성하였다.

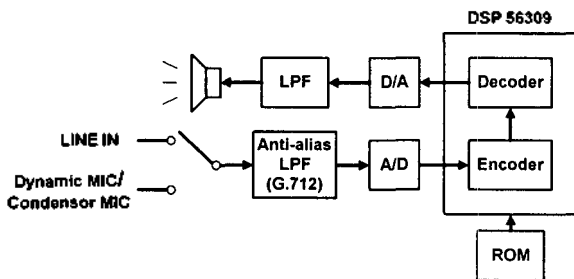


그림 3. 음성 코더의 실시간 동작시험 환경

Anti-aliasing 목적으로 ITU-T G.712 규격에 따라 6차 Chebychev [6] 저주파대역 통과 필터를 설계하여 적용하였으며, D/A 변환기를 통과시킨 후 출력에서 나오는 입력시에 적용한 동일 필터를 reconstruction 필터 목적으로 적용시켰다. ROM에 음성코더의 알고리즘을 프로그램하여, 독립적으로 DSP를 구동시킬 수 있는 환경을 구성하였다. 다양한 형태의 입력 음성에 대하여서 toll-quality에 달하는 수준의 음성 압축 및 재생을 확인하였다. ITU-T에서 제공하는 11개

의 시험벡터를 입력으로 하여 16bit 기반으로 설정된 DSP에서 코더 알고리즘을 수행하여 얻은 결과와 host PC에서 C언어 레벨 시뮬레이션 하여 얻은 기준 결과를 단계별로 비교하여 코딩 레벨에서 최적화시켰으며, 모든 시험벡터의 패턴에 대해 올바른 출력값을 얻어 지는 것을 확인하였다.

### 6. 결론

본 논문에서는 DSP의 구조적 장점인 pipeline과 parallel move등을 활용한 최적화를 통하여 MIPS를 효율적으로 개선함으로써 ITU 표준인 G.723.1 음성 부호기를 약 26 MIPS로 구현함을 보였다. ITU-T에서 제공하는 모든 시험 벡터를 입력으로 하여 PC상에서의 C 언어 레벨의 시뮬레이션 결과와 bit 단위 비교를 통하여 구현한 DSP 음성코더의 동작을 검증하였으며, 또한 실시간 동작시험 환경을 구성하고 이를 이용하여 다양한 형태의 입력 음성에 대하여서 toll-quality에 달하는 수준의 음성 압축 및 재생을 확인하였다.

본 논문은 전자통신연구원의 연구비 지원에 의해 수행한 연구 내용임

### 참고 문헌

- [1] B. Madhukar and D.G. Jerry "Speech Coding in Mobile Radio Communications," *proc. IEEE*, vol.82, pp.900-918, June 1994.
- [2] A. M. Kondoz, *Digital Speech: Coding for Low Bit Rate Communications Systems*. Chichester, West Sussex, England: Wiley, 1994
- [3] A. Gersho, "Advances in speech and audio compression," *proc. IEEE*, vol.82, pp.900-918, June 1994.
- [4] A. S. Spanias, "Speech coding: A tutorial review," *proc. IEEE*, vol.82, pp.1541-1582, Oct. 1994
- [5] Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/sec, ITU-T Recommendation G.723.1, Mar. 1996.
- [6] R. W. schaffer and L. R. Rabiner, "Digital processing of speech signals," *proc. IEEE*, vol.63, pp.662-677, Apr. 1978.
- [7] N. S. Jayant and P. Noll, *Digital coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.